UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/787,226	02/27/2004	Ryan Mason	049051-0222	4844

31824   7590   03/17/2009
MCDERMOTT WILL & EMERY LLP
18191 VON KARMAN AVE.
SUITE 500
IRVINE, CA 92612-7108

EXAMINER
BELANI, KISHIN G

ART UNIT	PAPER NUMBER
2443	

MAIL DATE	DELIVERY MODE
03/17/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
	10/787,226	MASON ET AL.
Office Action Summary	Examiner	Art Unit	
	KISHIN G. BELANI	2443	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>22 December 2008</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1,3-7,9-12,17,19-21 and 23-32</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1,3-7,9-12,17,19-21 and 23-32</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

This action is in response to Applicant's amendment filed on 12/22/2008.

Independent claims 1, 7, 17 and 21 and dependent claims 26, 29 and 30 have been

amended. Claims 2, 8, 13-16, 18 and 22 have been cancelled. Claims 1, 3-7, 9-13,

17, 19-21 and 23-32 are now pending in the present application. The applicants'

amendments to claims are shown in **bold and italics**, and the examiner's response to

claim amendments is shown in **bold** in this office action. **This Action is made FINAL.**

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth
> in section 102 of this title, if the differences between the subject matter sought to be patented and the prior
> art are such that the subject matter as a whole would have been obvious at the time the invention was
> made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall
> not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148

USPQ 459 (1966), that are applied for establishing a background for determining

obviousness under 35 U.S.C. 103(a) are summarized as follows:

1.   Determining the scope and contents of the prior art.
2.   Ascertaining the differences between the prior art and the claims at issue.
3.   Resolving the level of ordinary skill in the pertinent art.
4.   Considering objective evidence present in the application indicating
      obviousness
      or nonobviousness.

This application currently names joint inventors. In considering patentability of

the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g)

prior art under 35 U.S.C. 103(a).

Claims 1, 7, 17 and 21 are rejected under 35 U.S.C. 103(a) as being

unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of

Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1) and

further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1).**

Consider **claim 1**, Falcon et al. show and disclose a user interface for managing

a connection between a remote computing device and a local computing device (Fig. 6

that shows a user interface for setting up network and Internet connections; Fig. 7 that

shows a second interface to manage connections by setting different configuration

options for a connection; column 2, lines 7-9 disclose the same details), comprising:

a desktop *at the remote computing device*, wherein the desktop is operative to

display at least a first connection icon directly on the desktop, the first connection icon

representing a first connection between the remote computing device and a first local

computing device (Fig. 6 that shows **a window over** a desktop with a plurality of

connection icons named "Office", "Work From Home" and "MSN" to select from, wherein

each connection icon (under the "Connector Name" heading) represents a connection

between a server (a first local computing device) and a client/user computer (the remote

computing device); column 6, lines 29-43 describe the connection interface in more

details),

wherein a user can either select the first connection icon or an active area on the

desktop (Fig. 6 that further shows a "New Connector" active area on the desktop

window to define a new connection or to select one of the previously defined

connection (three of which are shown); column 6, lines 29-38 describe the same

details),

wherein selecting the first connection icon allows a first connection represented by the

first connection icon to become modifiable to alter the first connection (Fig. 7 that shows

a user interface for receiving configuration information from a user for a selected

connection (e.g. configuration information for the "Office" connection shown in Fig. 6);

further showing different tabs that allow a user to modify the configuration settings of the

selected connection; column 7, lines 1-27 disclose the details of the connection

properties that may be modified for each one of the tabs shown in Fig. 7),

wherein selecting the active area allows a new connection window to appear (column 6,

lines 34-38 which disclose using the "New Connection Wizard" by clicking on the active

area labeled "New Connector" to initiate a new connection, then configuring it by

supplying property values for the new connection object in Fig. 7) and, upon designating

a new connection, allows a second connection icon to be displayed directly on the

desktop (Fig. 6 that shows three different connections on the desktop **window** that were

created by the new connection wizard, then displayed as icons in the desktop **window**;

column 6, lines 29-34 describe the same details),

wherein the second connection icon represents a second connection different from the

first connection, between the remote computing device and a second local computing

device (Fig. 6 that shows a first connection "Office" icon, connecting a client's

computer/workstation with the server on an office LAN, and a second connection "Work

From Home" icon, connecting a client's computer/laptop at home with the server on the

office LAN network).

However, Falcon et al. do not specifically disclose that the first connection icon is

for a first application and the second connection icon is for a second application,

wherein the first application is different from the second application*; and wherein the*

desktop is operative to display at least a first application icon directly on the

desktop at the remote computing device, wherein the first application icon

represents an application available for execution on the first local computing

device.

In the same field of endeavor, Coulthard et al. show and disclose the claimed

user interface, wherein a first connection is for a first application and the second

connection is for a second application, wherein the first application is different from the

second application*; and wherein the desktop is operative to display at least a first*

application icon directly on the desktop at the remote computing device, wherein

the first application icon represents an application available for execution on the

first local computing device (Fig. 11, that shows three different connections 1111-

1113 between a Developer 1 and three Remote Systems 1120, 1122 and 1124, wherein

connection 1111 **(the first connection)** provides access to Tool A (application 1130) **to**

be executed on the "Remote System 1", i.e. 1120 (corresponding to the first local

computing device, a first server) and connection 1112 **(the second connection)**

provides access to Tool C (application 1150) **to be executed on the "Remote System**

2", i.e. 1122 (corresponding to the second local computing device, a second

server); paragraph 0099 describes the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a user interface on a desktop, wherein a first

connection is for a first application and the second connection is for a second

application, and wherein the first application is different from the second application,

and wherein the desktop is operative to display at least a first application icon directly

on the desktop at the remote computing device, wherein the first application icon

represents an application available for execution on the first local computing device, as

taught by Coulthard et al., in the user interface of Falcon et al., so as to provide a user a

graphical interface to set up and manage network connections based on the needed

applications.

Although the combined references of Falcon et al. and Coulthard et al.

disclose each and every element of claim 1; the applicants have argued that

having a connection icon on a desktop is not equivalent to having a connection

icon in a window which is displayed on a desktop. Although dragging and

dropping an icon from a displayed window to a desktop is a common and

obvious technique in a Windows operating system, the examiner has included the

additional reference of Handel et al. specifically to show that <u>placing a connection</u>

<u>icon on a desktop</u> has indeed been previously disclosed, as described in column

30, lines 24-32 of the Handel et al. reference.

        Consider **claim 7**, Falcon et al. show and disclose a method for managing a

connection between a local computing device and a remote computing device using a

user interface (Fig. 6 that shows a user interface for setting up network and Internet

connections; Fig. 7 that shows a second interface to manage connections by setting

different configuration options for a connection; column 2, lines 7-9 disclose the same

details), comprising the steps of:

displaying a desktop ***at the remote computing device*** (Fig. 6 that displays a desktop

with a user interface for setting up network and Internet connections **at the remote**

computing device);

displaying at least a first connection icon directly on the desktop, the first connection

icon representing a first connection between the remote computing device and a first

local computing device (Fig. 6 that shows a **window over a** desktop with a plurality of

connection icons named "Office", "Work From Home" and "MSN" to select from, wherein

each connection icon (under the "Connector Name" heading) represents a connection

between a server (a first local computing device) and a client/user computer (the remote

computing device); column 6, lines 29-43 describe the connection interface in more

details);

receiving a user selection of the first connection icon, wherein the user selection of the

first connection icon allows a first connection represented by the first connection icon to

become modifiable to alter the first connection (Fig. 7 that shows a user interface for

receiving configuration information from a user for a selected connection (e.g.

configuration information for the "Office" connection shown in Fig. 6); further showing

different tabs that allow a user to modify the configuration settings of the selected

connection; column 7, lines 1-27 disclose the details of the connection properties that

may be modified for each one of the tabs shown in Fig. 7);

receiving a user selection of an active area of the desktop (Fig. 6 that further shows a

"New Connector" active area on the desktop **window** to define a new connection;

column 6, lines 29-38 describe the same details);

wherein the user selection of the active area allows a second connection icon to be

displayed directly on the desktop, wherein the second connection icon represents a

second connection different than the first connection (Fig. 6 that shows a second

connection "Work From Home" icon, connecting a client's computer/laptop at home with

the server on the Office LAN network, which is different than a first connection (shown

as "Office" in Fig. 6)).

However, Falcon et al. do not specifically disclose that the first connection icon is

for a first application and the second connection icon is for a second application*; and*

displaying at least a first application icon on the desktop at the remote computing

device, wherein the first application icon represents an application available for execution on the first local computing device.

In the same field of endeavor, Coulthard et al. show and disclose the claimed method, wherein a first connection is for a first application and the second connection is for a second application*, and displaying at least a first application icon on the desktop at the remote computing device, wherein the first application icon represents an application available for execution on the first local computing device* (Fig. 11, that shows three different connections 1111-1113 between a Developer 1 and three Remote Systems 1120, 1122 and 1124, wherein connection 1111 **(the first connection)** provides access to Tool A (application 1130) **to be executed on the "Remote System 1", i.e. 1120 (corresponding to the first local computing device, a first server)** and connection 1112 **(the second connection)** provides access to Tool C (application 1150) **to be executed on the "Remote System 2", i.e. 1122 (corresponding to the second local computing device, a second server)**; paragraph 0099 describes the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method, wherein a first connection is for a first application and the second connection is for a second application, and displaying at least a first application icon on the desktop at the remote computing device, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Coulthard et al., in the method of Falcon et al., so as to

provide a user a graphical interface to set up and manage network connections based on the needed applications.

Although the combined references of Falcon et al. and Coulthard et al. disclose each and every element of claim 1; the applicants have argued that having a connection icon on a desktop is not equivalent to having a connection icon in a window which is displayed on a desktop. Although dragging and dropping an icon from a displayed window to a desktop is a common and obvious technique in a Windows operating system, the examiner has included the additional reference of Handel et al. specifically to show that <u>placing a connection icon on a desktop</u> has indeed been previously disclosed, as described in column 30, lines 24-32 of the Handel et al. reference.

Consider **claim 17**, Falcon et al. disclose a computer-executable program code stored on a computer readable medium for managing a connection between a local computing device and a remote computing device using a user interface (claim 11; Fig. 6 that shows a user interface for setting up network and Internet connections; Fig. 7 that shows a second interface to manage connections by setting different configuration options for a connection; column 2, lines 7-9 disclose the same details), the computer-executable program code comprising:

code for displaying a desktop *at the remote computing device* (Fig. 6 that displays a desktop with a user interface for setting up network and Internet connections **at the remote computing device**);

code for displaying at least a first connection icon directly on the desktop, the first

connection icon representing a first connection between the remote computing device

and a first local computing device (Fig. 6 that shows **a window over** a desktop with a

plurality of connection icons named "Office", "Work From Home" and "MSN" to select

from, wherein each connection icon (under the "Connector Name" heading) represents

a connection between a server (a first local computing device) and a client/user

computer (the remote computing device); column 6, lines 29-43 describe the connection

interface in more details);

code for receiving a user selection of the first connection icon, wherein the user

selection of the first connection icon allows a first connection represented by the first

connection icon to become modifiable to alter the first connection (Fig. 7 that shows a

user interface for receiving configuration information from a user for a selected

connection (e.g. configuration information for the "Office" connection shown in Fig. 6);

further showing different tabs that allow a user to modify the configuration settings of the

selected connection; column 7, lines 1-27 disclose the details of the connection

properties that may be modified for each one of the tabs shown in Fig. 7);

code for receiving a user selection of an active area of the desktop (Fig. 6 that further

shows a "New Connector" active area on the desktop to define a new connection;

column 6, lines 29-38 describe the same details),

wherein the user selection of the active area allows a second connection icon to be

displayed directly on the desktop (Fig. 6 that shows three different connections on the

desktop **window** that were created by the new connection wizard, then displayed as

icons on the desktop **window**; column 6, lines 29-34 describe the same details),

wherein the second connection icon represents a second connection different than the

first connection (Fig. 6 that shows a first connection "Office" icon, connecting a client's

computer/workstation with the server on an office LAN, and a second connection "Work

From Home" icon, connecting a client's computer/laptop at home with the server on the

Office LAN network).

However, Falcon et al. do not specifically disclose that the first connection icon is

for a first application and the second connection icon is for a second application*, and*

code for displaying at least a first application icon on the desktop at the remote

computing device, wherein the first application icon represents an application

available for execution on the first local computing device.

In the same field of endeavor, Coulthard et al. show and disclose the claimed

computer-executable program code, wherein a first connection is for a first application

and the second connection is for a second application, *and code for displaying at*

least a first application icon on the desktop at the remote computing device,

wherein the first application icon represents an application available for

execution on the first local computing device (Fig. 11, that shows three different

connections 1111-1113 between a Developer 1 and three Remote Systems 1120, 1122

and 1124, wherein connection 1111 **(the first connection)** provides access to Tool A

(application 1130) **to be executed on the "Remote System 1", i.e. 1120**

(corresponding to the first local computing device, a first server) and connection

1112 **(the second connection)** provides access to Tool C (application 1150) **to be executed on the "Remote System 2", i.e. 1122 (corresponding to the second local computing device, a second server)**; paragraph 0099 describes the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a computer-executable program code stored on a computer-readable medium, wherein a first connection is for a first application and the second connection is for a second application, and code for displaying at least a first application icon on the desktop at the remote computing device, wherein the first application icon represents an application available for execution on the first local computing device, as taught by Coulthard et al., in the computer-executable program code of Falcon et al., so as to provide a user with the executable program code to set up and manage network connections based on the needed applications.

Although the combined references of Falcon et al. and Coulthard et al. disclose each and every element of claim 1; the applicants have argued that having a connection icon on a desktop is not equivalent to having a connection icon in a window which is displayed on a desktop. Although dragging and dropping an icon from a displayed window to a desktop is a common and obvious technique in a Windows operating system, the examiner has included the additional reference of Handel et al. specifically to show that <u>placing a connection icon on a desktop</u> has indeed been previously disclosed, as described in column 30, lines 24-32 of the Handel et al. reference.

Consider **claim 21**, Falcon et al. show and disclose a programmed computer

apparatus for managing a connection between a local computing device and a remote

computing device using a user interface (Fig. 6 that shows a computer desktop with a

user interface for setting up network and Internet connections; Fig. 7 that shows a

computer desktop with a user interface to manage connections by setting different

configuration options for a connection; column 2, lines 7-9 disclose the same details),

said programmed computer apparatus comprising:

means for displaying a desktop *at the remote computing device* (Fig. 6 that displays a

computer desktop with a user interface for setting up network and Internet connections

at the remote computing device),

means for displaying at least a first connection icon directly on the desktop, the first

connection icon representing a first connection between the remote computing device

and a first local computing device (Fig. 6 that shows a **window over** a desktop on a

computer with a plurality of connection icons named "Office", "Work From Home" and

"MSN" to select from, wherein each connection icon (under the "Connector Name"

heading) represents a connection between a server (a first local computing device) and

a client/user computer (the remote computing device); column 6, lines 29-43 describe

the claimed apparatus in more details);

means for receiving a user selection of the first connection icon, wherein the user

selection of the first connection icon allows a first connection represented by the first

connection icon to become modifiable to alter the first connection (Fig. 7 that shows a

user interface for receiving configuration information from a user for a selected

connection (e.g. configuration information for the "Office" connection shown in Fig. 6);

further showing different tabs that allow a user to modify the configuration settings of the

selected connection; column 7, lines 1-27 disclose the details of the connection

properties that may be modified for each one of the tabs shown in Fig. 7);

means for receiving a user selection of an active area of the desktop, wherein the user

selection of the active area allows a second connection icon for a second application to

be displayed directly on the desktop (Fig. 6 that shows a "New Connector" active area

on the desktop **window** to define a new connection; Fig. 6 further shows three different

connections on the desktop **window** that were created by the new connection wizard,

then displayed as icons on the desktop **window**; column 6, lines 29-38 describe the

same details);

wherein the second connection icon represents a second connection different than the

first connection (Fig. 6 that shows a first connection "Office" icon, connecting a client's

computer/workstation with the server on an office LAN, and a second connection "Work

From Home" icon, connecting a client's computer/laptop at home with the server on the

Office LAN network).

However, Falcon et al. do not specifically disclose that the first connection icon is

for a first application and the second connection icon is for a second application, **and**

means for displaying at least a first application icon on the desktop at the remote

computing device, wherein the first application icon represents an application

available for execution on the first local computing device.

In the same field of endeavor, Coulthard et al. show and disclose the claimed

programmed computer apparatus, wherein a first connection is for a first application and

the second connection is for a second application, *and means for displaying at least*

a first application icon on the desktop at the remote computing device, wherein

the first application icon represents an application available for execution on the

first local computing device (Fig. 11, that shows three different connections 1111-

1113 between a Developer 1 and three Remote Systems 1120, 1122 and 1124, wherein

connection 1111 **(the first connection)** provides access to Tool A (application 1130) **to**

be executed on the "Remote System 1", i.e. 1120 (corresponding to the first local

computing device, a first server) and connection 1112 **(the second connection)**

provides access to Tool C (application 1150) **to be executed on the "Remote System**

2", i.e. 1122 (corresponding to the second local computing device, a second

server); paragraph 0099 describes the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a programmed computer apparatus,

wherein a first connection is for a first application and the second connection is for a

second application, and means for displaying at least a first application icon on the

desktop at the remote computing device, wherein the first application icon represents an

application available for execution on the first local computing device, as taught by

Coulthard et al., in the programmed computer apparatus of Falcon et al., so as to

provide a user an apparatus to set up and manage network connections based on the

needed applications.

Although the combined references of Falcon et al. and Coulthard et al.

disclose each and every element of claim 1; the applicants have argued that

having a connection icon on a desktop is not equivalent to having a connection

icon in a window which is displayed on a desktop. Although dragging and

dropping an icon from a displayed window to a desktop is a common and

obvious technique in a Windows operating system, the examiner has included the

additional reference of Handel et al. specifically to show that <u>placing a connection</u>

<u>icon on a desktop</u> has indeed been previously disclosed, as described in column

30, lines 24-32 of the Handel et al. reference.

Claims 3, 9, 19 and 23 are rejected under 35 U.S.C. 103(a) as being

unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of

Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1) and

further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1)** and further in

view of **Perholtz et al. (U.S. Patent Application Publication # 2002/0091850 A1).**

Consider **claim 3**, and **as it applies to claim 1 above**, Falcon et al., as modified

by Coulthard et al. and Handel et al., show and disclose a user interface for managing a

connection between a remote computing device and a local computing device, except

further comprising a keystroke management window, wherein the keystroke

management window is user modifiable to accept a local keystroke management

setting, wherein if the local keystroke management setting is enabled, a keystroke is

processed at the remote computing device, and wherein if the local keystroke
management setting is disabled, the keystroke is processed at the first local computing
device.

In the same field of endeavor, Perholtz et al. disclose a user interface, further
comprising a keystroke management window, wherein the keystroke management is
user modifiable to accept a local keystroke management setting, wherein if the local
keystroke management setting is enabled, a keystroke is processed at the remote
computing device, and wherein if the local keystroke management setting is disabled,
the keystroke is processed at the first local computing device (Flowchart of Fig. 7G,
decision block 759 that checks for use of "Hot Keys"; paragraph 0288, lines 1-16 that
disclose the use of "Hot Keys" for redirecting remote client's input keystrokes/mouse
data to the local server and means to return back to the remote client's normal mode of
operation by tapping the left shift key three times within 2 seconds; although no window
is shown for selecting an option to make hot key sequences effective either at a local
computing device or at a remote computing device, the paragraph does mention
selecting from a menu, either Remote PC mode or Host mode. Based on the user
selection, the hot keys are either applicable at the remote computing device, or at the
local computing device. Therefore, the examiner has taken the official notice that the
use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface,
as is evident when copying a paragraph from one document and pasting it into another
document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a
pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a user interface, further comprising a keystroke management window, wherein the keystroke management is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device, as taught by Perholtz et al., in the user interface of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to provide a user ability to use hot keys to execute applications at the local computing device as well as at the remote computing device, and be able to easily switch between them.

Consider **claim 9**, and **as it applies to claim 7 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., show and disclose the claimed method for managing a connection between a remote computing device and a local computing device, except further comprising the step of displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device.

In the same field of endeavor, Perholtz et al. disclose the claimed method, further comprising the step of displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Flowchart of Fig. 7G, decision block 759 that checks for use of "Hot Keys"; paragraph 0288, lines 1-16 that disclose the use of "Hot Keys" for redirecting remote client's input keystrokes/mouse data to the local server and means to return back to the remote client's normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method, further comprising the step of

displaying a keystroke management window, wherein the keystroke management

window is user modifiable to accept a local keystroke management setting, wherein if

the local keystroke management setting is enabled, a keystroke is processed at the

remote computing device, and wherein if the local keystroke management setting is

disabled, the keystroke is processed at the first local computing device, as taught by

Perholtz et al., in the method of Falcon et al., as modified by Coulthard et al. and

Handel et al., so as to provide a user ability to use hot keys to execute applications at

the local computing device as well as at the remote computing device, and be able to

easily switch between them.

Consider **claim 19**, and **as it applies to claim 17 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., show and disclose the claimed computer-

executable program code, except code for displaying a keystroke management window,

wherein the keystroke management window is user modifiable to accept a local

keystroke management setting, wherein if the local keystroke management setting is

enabled, a keystroke is processed at the remote computing device, and wherein if the

local keystroke management setting is disabled, the keystroke is processed at the first

local computing device.

In the same field of endeavor, Perholtz et al. disclose the claimed computer-

executable program code, comprising code for displaying a keystroke management

window, wherein the keystroke management window is user modifiable to accept a local

keystroke management setting, wherein if the local keystroke management setting is

enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke management setting is disabled, the keystroke is processed at the first local computing device (Claims 1-5; Flowchart of Fig. 7G, decision block 759 that checks for use of "Hot Keys"; paragraph 0288, lines 1-16 that disclose the use of "Hot Keys" for redirecting remote client's input keystrokes/mouse data to the local server and means to return back to the remote client's normal mode of operation by tapping the left shift key three times within 2 seconds; although no window is shown for selecting an option to make hot key sequences effective either at a local computing device or at a remote computing device, the paragraph does mention selecting from a menu, either Remote PC mode or Host mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pulldown menu (GUI) or a toolbar icon to copy the paragraph).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a computer-executable program code, comprising code for displaying a keystroke management window, wherein the keystroke management window is user modifiable to accept a local keystroke management setting, wherein if the local keystroke management setting is enabled, a keystroke is processed at the remote computing device, and wherein if the local keystroke

management setting is disabled, the keystroke is processed at the first local computing

device, as taught by Perholtz et al., in the computer-executable program code of Falcon

et al., as modified by Coulthard et al. and Handel et al., so as to provide a user ability to

use hot keys to execute applications at the local computing device as well as at the

remote computing device, and be able to easily switch between them.

Consider **claim 23**, and **as it applies to claim 21 above**, Falcon et al., as

modified by Coulthard et al., Handel et al., and Perholtz et al., further show and disclose

a programmed computer apparatus for managing a connection between a local

computing device and a remote computing device using a user interface, comprising

means for displaying a keystroke management window, wherein the keystroke

management window is user modifiable to accept a local keystroke management

setting, wherein if the local keystroke management setting is enabled, a keystroke is

processed at the remote computing device, and wherein if the local keystroke

management setting is disabled, the keystroke is processed at the first local computing

device (Flowchart of Fig. 7G, decision block 759 that checks for use of "Hot Keys";

paragraph 0288, lines 1-16 that disclose the use of "Hot Keys" for redirecting remote

client's input keystrokes/mouse data to the local server and means to return back to the

remote client's normal mode of operation by tapping the left shift key three times within

2 seconds; although no window is shown for selecting an option to make hot key

sequences effective either at a local computing device or at a remote computing device,

the paragraph does mention selecting from a menu, either Remote PC mode or Host

mode. Based on the user selection, the hot keys are either applicable at the remote computing device, or at the local computing device. Therefore, the examiner has taken the official notice that the use of keystrokes achieves the same purpose as the mouse clicks on a GUI interface, as is evident when copying a paragraph from one document and pasting it into another document. One may use Ctrl-c keyboard keys to copy a selected paragraph or use a pull-down menu (GUI) or a toolbar icon to copy the paragraph).

Claims 4, 10, 25, 27, 28 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of **Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1)** and further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1)** and further in view of **Beadle et al. (U.S. Patent Publication # 7,039,709 B1).**

Consider **claim 4**, and **as it applies to claim 1 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., disclose the claimed user interface, except wherein the first connection icon and the second connection icon each includes a priority.

In the same field of endeavor, Beadle et al. disclose a user interface for managing a connection between a remote computing device and a local computing device, wherein the first connection icon and the second connection icon each includes a priority (in Beadle et al. reference, Fig. 5A, "Select Default Server" block 507,

"Override Defaults" block 511, and "Update Settings" button 515 that enable a user to set priorities in selecting different connections and other options; Fig. 6A that lists some of the options 601 that can be assigned priority values to arrive at the relative ratings 605; column 8, lines 28-33 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a user interface, wherein the first connection icon and the second connection icon each includes a priority, as taught by Beadle et al., in the user interface of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to allow users to assign different priorities to defined connections.

Consider **claim 10**, and **as it applies to claim 7 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., disclose the claimed method, except wherein the first connection icon and the second connection icon each includes a priority.

In the same field of endeavor, Beadle et al. disclose a method for managing a connection between a local computing device and a remote computing device, using a user interface, wherein the first connection icon and the second connection icon each includes a priority (in Beadle et al. reference, Fig. 5A, "Select Default Server" block 507, "Override Defaults" block 511, and "Update Settings" button 515 that enable a user to set priorities in selecting different connections and other options; Fig. 6A that lists some of the options 601 that can be assigned priority values to arrive at the relative ratings 605; column 8, lines 28-33 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a method for a user interface, wherein the first connection icon and the second connection icon each includes a priority, as taught by Beadle et al., in the method of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to allow users to assign different priorities to defined connections.

Consider **claim 25**, and **as it applies to claim 1 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., further disclose the claimed user interface for managing a connection between a remote computing device and a local computing device, wherein selecting the first connection icon allows the user to edit or delete the first connection (in Falcon et al. reference, Fig. 7 which shows a second interface to manage connections by setting different configuration options for a connection; column 2, lines 7-9 disclose the same details).

However, Falcon et al., as modified by Coulthard et al. and Handel et al., do not specifically disclose the user interface wherein the remote computing device is a thin client, and wherein the user interface is to be displayed at the thin client.

In the same field of endeavor, Beadle et al. disclose the claimed user interface for managing a connection between a remote computing device and a local computing device, wherein the remote computing device is a thin client (column 1, lines 32-34 which disclose that clients can be "dumber" systems (thin clients) adapted for limited use with a network); and

wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that

disclose a graphical user interface for receiving user selection at the remote thin client,

and a connection utility for connecting the client with a selected local server).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a user interface, wherein the remote

computing device is a thin client, and wherein the user interface is to be displayed at the

thin client, as taught by Beadle et al., in the user interface of Falcon et al., as modified

by Coulthard et al. and Handel et al., so as to provide support for connection

management to clients with thin remote devices.

Consider **claim 27**, and **as it applies to claim 17 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., disclose the claimed computer-

executable program code,

wherein selecting the first connection icon allows the user to edit or delete the first

connection (in Falcon et al. reference, Fig. 7 which shows a second interface to manage

connections by setting different configuration options for a connection; column 2, lines

7-9 disclose the same details); and

wherein the first application is different from the second application (in Coulthard et al.

reference, Fig. 11, that shows three different connections 1111-1113 between a

Developer 1 and three Remote Systems 1120, 1122 and 1124, wherein connection

1111 provides access to Tool A (application 1130) and connection 1112 provides

access to Tool C (application 1150); paragraph 0099 describes the same details).

However, Falcon et al., as modified by Coulthard et al. and Handel et al., do not specifically disclose that the remote computing device is a thin client, wherein the user interface is to be displayed at the thin client, and wherein the second connection is between the thin client and a second local computing device.

In the same field of endeavor, Beadle et al. disclose the claimed computer-executable program code for managing a connection between a remote computing device and a local computing device, wherein the remote computing device is a thin client (column 1, lines 32-34 which disclose that clients can be "dumber" systems (thin clients) adapted for limited use with a network);
wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that disclose a graphical user interface for receiving user selection at the remote thin client, and a connection utility for connecting the client with a selected local server); and wherein the second connection is between the thin client and a second local computing device (Fig. 10 that shows a second connection using modem transmission; column 10, lines 6-24 which disclose a first connection via satellite to a DirectPC application and a second modem connection to a server for a financial application).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide computer-executable program code, wherein the remote computing device is a thin client, wherein the user interface is to be displayed at the thin client, and wherein the second connection is between the thin client and a second local computing device, as taught by Beadle et al., in the computer-executable program code of Falcon et al., as modified by Coulthard et al. and Handel et

al., so as to provide support for connection management to clients with thin remote

devices.

Consider **claim 28**, and **as it applies to claim 21 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., disclose the claimed programmed

computer apparatus,

wherein the user selection of the first connection icon allows a user at the thin client to

edit or delete the first connection (in Falcon et al. reference, Fig. 7 which shows a

second interface to manage connections by setting different configuration options for a

connection; column 2, lines 7-9 disclose the same details); and wherein the first

application is different from the second application (in Coulthard et al. reference, Fig.

11, that shows three different connections 1111-1113 between a Developer 1 and three

Remote Systems 1120, 1122 and 1124, wherein connection 1111 provides access to

Tool A (application 1130) and connection 1112 provides access to Tool C (application

1150); paragraph 0099 describes the same details).

However, Falcon et al., as modified by Coulthard et al. and Handel et al., do not

specifically disclose that the remote computing device is a thin client, wherein the user

interface is to be displayed at the thin client, wherein the active area is an empty area,

and wherein the second connection is between the thin client and a second local

computing device.

In the same field of endeavor, Beadle et al., disclose the claimed programmed

computer apparatus, wherein the remote computing device is a thin client (in Beadle et

al. reference, claims 10-12; column 1, lines 32-34 which disclose that clients can be

"dumber" systems (thin clients) adapted for limited use with a network);

wherein the user interface is to be displayed at the thin client (column 2, lines 54-57 that

disclose a graphical user interface for receiving user selection at the remote thin client,

and a connection utility for connecting the client with a selected local server);

wherein the active area is an empty area (in Beadle et al. reference, Fig. 5A, desktop

area 507, 511 or 513 that does not include the first connection areas 505A-D (this

interpretation of empty area based on the applicants' definition of empty area in

paragraph 0075 of the application stating that empty area is where desktop is not

displaying first connection)); and

wherein the second connection is between the thin client and a second local computing

device (in Beadle et al. reference, Fig. 10 that shows a second connection using

modem transmission; column 10, lines 6-24 which disclose a first connection via

satellite to a DirectPC application and a second modem connection to a server for a

financial application).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a programmed computer apparatus,

wherein the remote computing device is a thin client, wherein the user interface is to be

displayed at the thin client, wherein the active area is an empty area, and wherein the

second connection is between the thin client and a second local computing device, as

taught by Beadle et al., in the programmed computer apparatus of Falcon et al., as

modified by Coulthard et al. and Handel et al., so as to provide support for connection

management to clients with thin remote devices.

Consider **claim 31**, and **as it applies to claim 17 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., disclose the claimed computer-

executable program code, except wherein the first connection icon and the second

connection icon each includes a priority.

In the same field of endeavor, Beadle et al. disclose computer-executable

program code, wherein the first connection icon and the second connection icon each

includes a priority (in Beadle et al. reference, claims 10-12; Fig. 5A, "Select Default

Server" block 507, "Override Defaults" block 511, and "Update Settings" button 515 that

enable a user to set priorities in selecting different connections and other options; Fig.

6A that lists some of the options 601 that can be assigned priority values to arrive at the

relative ratings 605; column 8, lines 28-33 that disclose the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide computer-executable program code,

wherein the first connection icon and the second connection icon each includes a

priority, as taught by Beadle et al., in the computer-executable program code of Falcon

et al., as modified by Coulthard et al. and Handel et al., so as to allow users to assign

different priorities to defined connections.

Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Falcon**

et al. (U.S. Patent Publication # 6,295,556 B1) in view of **Coulthard et al. (U.S.**

Patent Application Publication # 2004/0003371 A1) and further in view of **Handel et**

al. (U.S. Patent Publication # 7,076,504 B1) and further in view of **Lele (U.S. Patent**

Publication # 7,181,524 B1).

Consider **claim 5**, and **as it applies to claim 1 above**, Falcon et al., as modified

by Coulthard et al. and Handel et al., disclose the claimed user interface, except

wherein the priority is a failover order.

In the same field of endeavor, Lele discloses a user interface, wherein the priority

is a failover order (column 1, lines 21-27 that disclose a plurality of servers connected in

a server cluster to provide failover redundancy; Fig. 1, Rules block 154 and Selection

Algorithm block 155 that specify server selection criteria; thereby disclosing server

failover order that a user may specify as a priority option in the connection

management).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a user interface for managing a connection

between a remote computing device and a local computing device, wherein the priority

is a failover order, as taught by Lele, in the user interface of Falcon et al., as modified

by Coulthard et al. and Handel et al., so as to provide an alternate connection path to a

server, in case the selected server fails.

Claims 6, 12, 20 and 24 are rejected under 35 U.S.C. 103(a) as being

unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of

Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1) and

further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1)** and further in

view of **Ritchy et al. (U.S. Patent Application Publication # 2004/0183831 A1).**

Consider **claim 6**, and **as it applies to claim 1 above**, Falcon et al., as modified

by Coulthard et al. and Handel et al., disclose the claimed user interface, except further

comprising a desktop shell window, wherein the desktop shell window is modifiable at

run-time by the user at the remote computing device to accept a desktop shell setting,

wherein if the desktop shell setting is disabled, an alternate user interface is selected

and the user interface is disabled.

In the same field of endeavor, Ritchy et al. disclose a desktop window, wherein

the desktop shell window is modifiable at run-time by the user at the remote computing

device to accept a desktop shell setting, wherein if the desktop shell setting is disabled,

an alternate user interface is selected and the user interface is disabled (Fig. 9 that

shows a default desktop window and a pull-down to select alternate desktop shell if the

user so desires; paragraph 0049, lines 9-11 which disclose that different shells for the

desktop are selectable in the Property Editor window, and portal administrators and end

users can also change a desktop's shell, thereby disclosing that the desktop shell

window is modifiable at run-time by the user at the remote computing device to accept a

desktop shell setting; wherein if the desktop shell setting is disabled, an alternate user

interface is selected and the user interface is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a desktop shell window, wherein the

desktop shell window is modifiable at run-time by the user at the remote computing

device to accept a desktop shell setting, wherein if the desktop shell setting is disabled,

an alternate user interface is selected and the user interface is disabled, as taught by

Ritchy et al., in the user interface of Falcon et al., as modified by Coulthard et al. and

Handel et al., so as to provide multiple operating systems environments for the user to

choose from, based on user's preferences, on the same desktop.

Consider **claim 12**, and **as it applies to claim 7 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., disclose the claimed method, except

further comprising the steps of displaying a desktop shell window, wherein the desktop

shell window is modifiable at run-time by a user at the remote computing device to

accept a desktop shell setting; selecting an alternate user interface, if the desktop shell

setting is disabled; disabling the user interface, if the desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. disclose a desktop window, wherein

the desktop shell window is modifiable at run-time by a user at the remote computing

device to accept a desktop shell setting; selecting an alternate user interface, if the

desktop shell setting is disabled; disabling the user interface, if the desktop shell setting

is disabled (Fig. 9 that shows a default desktop window and a pull-down to select

alternate desktop shell if the user so desires; paragraph 0049, lines 9-11 which disclose

that different shells for the desktop are selectable in the Property Editor window, and

portal administrators and end users can also change a desktop's shell, thereby

disclosing that the desktop shell window is modifiable at run time by the user at the

remote computing device to accept a desktop shell setting, selecting an alternate user

interface, if the desktop shell setting is disabled, and disabling the user interface, if the

desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a desktop shell window, wherein the

desktop shell window is modifiable at run-time by a user at the remote computing device

to accept a desktop shell setting; selecting an alternate user interface, if the desktop

shell setting is disabled; disabling the user interface, if the desktop shell setting is

disabled, as taught by Ritchy et al., in the method of Falcon et al., as modified by

Coulthard et al. and Handel et al., so as to provide multiple operating systems

environments for the user to choose from, based on user's preferences, on the same

desktop.

Consider **claim 20**, and **as it applies to claim 17 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., disclose the claimed computer-

executable program code, except said program code comprising code for displaying a

desktop shell window, wherein the desktop shell window is modifiable at run-time by a

user at the remote computing device to accept a desktop shell setting; code for

selecting an alternate user interface, if the desktop shell setting is disabled; and code

for disabling the user interface, if the desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. disclose a computer-readable storage

medium with stored program code, said program comprising code for permitting the

computer to perform a step for displaying a desktop shell window, wherein the desktop

shell window is modifiable at run-time by a user at the remote computing device to

accept a desktop shell setting; a selecting step for selecting an alternate user interface,

if the desktop shell setting is disabled; a disabling step for disabling the user interface, if

the desktop shell setting is disabled (Claims 20-38, 60-80, and 101-120; that shows a

default desktop window with a user interface (pull-down) to select an alternate desktop

shell if the user so desires; paragraph 0049, lines 9-11 which disclose that different

shells for the desktop are selectable in the Property Editor window, and portal

administrators and end users can also change a desktop's shell, thereby disclosing that

the desktop shell window is modifiable at run-time by a user at the remote computing

device to accept a desktop shell setting, selecting an alternate user interface, if the

desktop shell setting is disabled, and disabling the user interface, if the desktop shell

setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a computer-readable storage medium with

stored program code for managing a connection between a local computing device and

a remote computing device, said program comprising code for permitting the computer

to perform a step for displaying a desktop shell window, wherein the desktop shell

window is modifiable at run-time by a user at the remote computing device to accept a

desktop shell setting; a selecting step for selecting an alternate user interface, if the

desktop shell setting is disabled; a disabling step for disabling the user interface, if the

desktop shell setting is disabled, as taught by Ritchy et al., in the computer-executable

program code of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to

provide a user ability to select any one of the many available desktop shells that is most

suited to the user.

Consider **claim 24**, and **as it applies to claim 21 above**, Falcon et al., as

modified by Coulthard et al. and Handel et al., show and disclose the claimed

programmed computer apparatus, except further comprising means for displaying a

desktop shell window, wherein the desktop shell window is modifiable at run-time by a

user at the remote computing device; means for selecting an alternate user interface, if

the desktop shell setting is disabled; and means for disabling the user interface, if the

desktop shell setting is disabled.

In the same field of endeavor, Ritchy et al. show and disclose the claimed

programmed computer apparatus, further comprising means for displaying a desktop

shell window, wherein the desktop shell window is modifiable at run time by a user at

the remote computing device; means for selecting an alternate user interface, if the

desktop shell setting is disabled; and means for disabling the user interface, if the

desktop shell setting is disabled (Fig. 9 that shows a default desktop window and a pull-

down to select alternate desktop shell if the user so desires; paragraph 0049, lines 9-11

which disclose that different shells for the desktop are selectable in the Property Editor window, and portal administrators and end users can also change a desktop's shell, thereby disclosing that the desktop shell window is modifiable at run-time by the user at the remote computing device to accept a desktop shell setting, selecting an alternate user interface, if the desktop shell setting is disabled, and disabling the improved user interface, if the desktop shell setting is disabled).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide means for displaying a desktop shell window, wherein the desktop shell window is modifiable at run time by a user at the remote computing device; means for selecting an alternate user interface, if the desktop shell setting is disabled; and means for disabling the user interface, if the desktop shell setting is disabled, as taught by Ritchy et al., in the programmed computer apparatus of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to provide a user ability to select any one of the many available desktop shells that is most suited to the user.

Claims 11 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of **Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1)** and further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1)** and further in view of **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** and further in view of **Lele (U.S. Patent Publication # 7,181,524 B1).**

Consider **claim 11**, and **as it applies to claim 10 above**, Falcon et al., as

modified by Coulthard et al., Handel et al., and Beadle et al., disclose the claimed

method, except wherein the priority is a failover order.

In the same field of endeavor, Lele discloses the claimed method, wherein the

priority is a failover order (column 1, lines 21-27 that disclose a plurality of servers

connected in a server cluster to provide failover redundancy; Fig. 1, Rules block 154

and Selection Algorithm block 155 that specify server selection criteria; thereby

disclosing server failover order that a user may specify as a priority option in the

connection management).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide a method for managing a connection

between a remote computing device and a local computing device using a user

interface, wherein the priority is a failover order, as taught by Lele, in the method of

Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., so as to

provide an alternate connection path to a server, in case the selected server fails.

Consider **claim 32**, and **as it applies to claim 31 above**, Falcon et al., as

modified by Coulthard et al., Handel et al., and Beadle et al., disclose the claimed

computer-executable program code, except wherein the priority is a failover order.

In the same field of endeavor, Lele discloses the claimed computer-executable

program code, wherein the priority is a failover order (column 1, lines 21-27 that

disclose a plurality of servers connected in a server cluster to provide failover

redundancy; Fig. 1, Rules block 154 and Selection Algorithm block 155 that specify

server selection criteria; thereby disclosing server failover order that a user may specify

as a priority option in the connection management).

Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to provide computer-executable program code for

managing a connection between a remote computing device and a local computing

device, wherein the priority is a failover order, as taught by Lele, in the computer-

executable program code of Falcon et al., as modified by Coulthard et al., Handel et al.,

and Beadle et al., so as to provide an alternate connection path to a server, in case the

selected server fails.

Claims 26, 29 and 30 are rejected under 35 U.S.C. 103(a) as being

unpatentable over **Falcon et al. (U.S. Patent Publication # 6,295,556 B1)** in view of

Coulthard et al. (U.S. Patent Application Publication # 2004/0003371 A1) and

further in view of **Handel et al. (U.S. Patent Publication # 7,076,504 B1)** and further in

view of **Beadle et al. (U.S. Patent Publication # 7,039,709 B1)** and further in view of

Muraca (U.S. Patent Application Publication # 2002/0055917 A1).

Consider **claim 26**, and **as it applies to claim 7 above**, Falcon et al., as

modified by Coulthard et al., and Handel et al., further show and disclose the claimed

method, wherein the user selection of the first connection icon allows a user to edit or

delete the first connection (in Falcon et al. reference, Fig. 7 which shows a second

interface to manage (edit or modify) connections by setting different configuration

options for a connection; column 2, lines 7-9 disclose the same details); and

wherein the first application is different from the second application (in Coulthard et al.

reference, Fig. 11, that shows three different connections 1111-1113 between a

Developer 1 and three Remote Systems 1120, 1122 and 1124, wherein connection

1111 provides access to Tool A (application 1130) and connection 1112 provides

access to Tool C (application 1150); paragraph 0099 describes the same details).

However, Falcon et al., as modified by Coulthard et al., and Handel et al., do not

specifically disclose that the remote computing device is a thin client, wherein the user

interface is to be displayed at the thin client, ***wherein the thin client's operating***

system is designed for generic remote computing devices, and wherein the second

connection is between the thin client and a second local computing device.

In the same field of endeavor, Beadle et al. disclose the claimed method, wherein

the remote computing device is a thin client (in Beadle et al. reference, column 1, lines

32-34 which disclose that clients can be "dumber" systems (thin clients) adapted for

limited use with a network);

wherein the user interface is to be displayed at the thin client (in Beadle et al. reference,

column 2, lines 54-57 that disclose a graphical user interface for receiving user

selection at the remote thin client, and a connection utility for connecting the client with

a selected local server); and

wherein the second connection is between the thin client and a second local computing device (in Beadle et al. reference, Fig. 10 that shows a second connection using modem transmission; column 10, lines 6-24 which disclose a first connection via satellite to a DirectPC application and a second modem connection to a server for a financial application).

However, Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., do not specifically disclose that *the thin client's operating system is designed for generic remote computing devices*.

In the same field of endeavor, Muraca discloses the claimed method *wherein the thin client's operating system is designed for generic remote computing devices* **(paragraphs 104 and 114 which disclose that thin clients such as PDA and Pocket PC use Windows CE operating system; paragraph 0197 which discloses streamlining the management and administration of access to external connections; Fig. 36 shows use of thin clients).**

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a thin client with generic operating system, as taught by Muraca, in the method of Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., so as to support connection management for thin clients that need a portable operating system, such as Microsoft Windows CE operating system.

Consider **claim 29**, and **as it applies to claim 1 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., show and disclose the claimed user interface, except wherein the remote computing device is a thin client, *and wherein the thin client's operating system is designed for generic remote computing devices.*

In the same field of endeavor, Beadle et al., show and disclose the claimed user interface, wherein the remote computing device is a thin client (column 1, lines 32-34 which disclose that clients can be "dumber" systems (thin clients) adapted for limited use with a network).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide the user interface for a thin client, as taught by Beadle et al., in the user interface of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to support connection management for thin clients.

However, Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., do not specifically disclose that *the thin client's operating system is designed for generic remote computing devices*.

In the same field of endeavor, Muraca discloses the claimed user interface *wherein the thin client's operating system is designed for generic remote computing devices* **(paragraphs 104 and 114 which disclose that thin clients such as PDA and Pocket PC use Windows CE operating system; paragraph 0197 which discloses streamlining the management and administration of access to external connections; Fig. 36 shows use of thin clients).**

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a thin client with a generic operating system for remote computing devices, as taught by Muraca, in the user interface of Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., so as to support connection management for thin clients that need a portable operating system, such as Microsoft® Windows® CE operating system.

Consider **claim 30**, and **as it applies to claim 17 above**, Falcon et al., as modified by Coulthard et al. and Handel et al., disclose the claimed computer-executable program code, except wherein the remote computing device is a thin client, *and wherein the thin client's operating system is designed for generic remote computing devices.*

In the same field of endeavor, Beadle et al., disclose the claimed computer-executable program code, wherein the remote computing device is a thin client (claims 10-12; column 1, lines 32-34 which disclose that clients can be "dumber" systems (thin clients) adapted for limited use with a network).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide the computer-executable program code for a thin client, as taught by Beadle et al., in the computer-executable program code of Falcon et al., as modified by Coulthard et al. and Handel et al., so as to support connection management for thin clients.

However, Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., do not specifically disclose that *the thin client's operating system is designed for generic remote computing devices*.

In the same field of endeavor, Muraca discloses the claimed computer-executable program code *wherein the thin client's operating system is designed for generic remote computing devices* **(claim 1; paragraphs 104 and 114 which disclose that thin clients such as PDA and Pocket PC use Windows CE operating system; paragraph 0197 which discloses streamlining the management and administration of access to external connections; Fig. 36 shows use of thin clients).**

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a generic operating system for remote computing devices, as taught by Muraca, in the computer-executable program code of Falcon et al., as modified by Coulthard et al., Handel et al., and Beadle et al., so as to support connection management for thin clients that need a portable operating system, such as Microsoft® Windows® CE operating system.

Response to Arguments

Applicant's arguments with respect to claims 1, 3-7, 9-12, 17, 19-21 and 23-32 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in

this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the date of this final action.

Any response to this Office Action should be **faxed to** (571) 273-8300 **or mailed**

to:

> Commissioner for Patents
> P.O. Box 1450
> Alexandria, VA 22313-1450
>
> Art Unit: 2443

Hand-delivered responses should be brought to

> Customer Service Window
> Randolph Building
> 401 Dulany Street
> Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the

Examiner should be directed to Kishin G. Belani whose telephone number is (571) 270-

1768. The Examiner can normally be reached on Monday-Friday from 6:00 am to 5:00

pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's

supervisor, Tonia Dollinger can be reached on (571) 272-4170. The fax phone number

for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for published

applications may be obtained from either Private PAIR or Public PAIR. Status

information for unpublished applications is available through Private PAIR only. For

more information about the PAIR system, see http://pair-direct.uspto.gov. Should you

have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll-free) or 703-305-3028.

Any inquiry of a general nature or relating to the status of this application or

proceeding should be directed to the receptionist/customer service whose telephone

number is (571) 272-0800.

/K. G. B./
Examiner, Art Unit 2443

March 12, 2009

/George C Neurauter, Jr./
Primary Examiner, Art Unit 2443